

Operación con variables que cuentan en la base de establecimientos y ofertas educativas.

ANEXO I

Script para RStudio con el cálculo de variables que cuentan

#####

#Generación base Padrón por oferta para generar variables que cuentan:

#####

#Cargar librerías

library(dplyr)

library(tidyverse)

library(haven)

library(xlsx)

library(openxlsx)

library(stringr)

#Llamo el DF del padrón con OA exportado y armado

#Lo que aparece entre comillas puede variar según el nombre de tu archivo

Padrón_OA <- read.xlsx ("Ruta/padron_31_03_25_oferta.xlsx")

#Primero armamos nivtip, variable de combinación de nivel y modalidad para contar UE y US.

Padrón_OA <- Padrón_OA %>%

arrange(cueanexo,

as.numeric(c_oferta))

```

Padrón_OA <- Padrón_OA %>%
  mutate(nivtip = case_when(
    c_oferta %in% 100:101 ~ 01,
    c_oferta %in% c(102, 105) ~ 03,
    c_oferta %in% c(109, 110, 111) ~ 06,
    c_oferta == 115 ~ 10,
    c_oferta %in% 116:119 ~ 99,
    c_oferta == 120 ~ 99,
    c_oferta %in% 121:122 ~ 02,
    c_oferta == 126 ~ 04,
    c_oferta %in% c(130, 132) ~ 07,
    c_oferta %in% 134:136 ~ 99,
    c_oferta == 138 ~ 99,
    c_oferta == 140 ~ 05,
    c_oferta == 144 ~ 08,
    c_oferta == 146 ~ 99,
    c_oferta == 161 ~ 08,
    c_oferta %in% c(170, 171) ~ 99,
    c_oferta %in% c(152, 155, 153, 156, 154, 159) ~ 99,
    TRUE ~ NA_real_ # Para cualquier otro valor que no esté especificado
  ))

```

#Probamos los valores de nivtip y consultamos que no hayan valores perdidos

#Consulta lógica

```
table(Padrón_OA$nivtip)
```

#Consulta numérica

```
any(is.na(Padrón_OA$nivtip))
```

#Contemos n_establecimientos

```

Padrón_OA <- Padrón_OA %>%
  arrange(cue, anexo, nivtip) %>%
  mutate(total_establecimientos = ifelse(row_number() == 1, 1,
                                          ifelse(cue !=
                                                  lag(cue, default = first(cue)), 1, 0)))

```

#Contemos n_localizaciones

```
Padrón_OA <- Padrón_OA %>%  
  arrange(cue, anexo, nivtip) %>%  
  mutate(total_localizaciones = ifelse(row_number() == 1, 1,  
                                       ifelse(cueanexo !=  
                                              lag(cueanexo, default = first(cueanexo)), 1, 0)))
```

#Contemos n_unidades_educativas

#Primero hay que ordenar por las variables que necesitamos en cada caso.

#En el caso de UE necesitamos CUE, nivtip, modalidad

```
Padrón_OA <- Padrón_OA %>%  
  arrange(cue, nivtip, c_modalidad1) %>%  
  mutate(total_ue = ifelse(row_number() == 1, 1,  
                           ifelse(cue != lag(cue, default = first(cue)) |  
                                nivtip != lag(nivtip, default = first(nivtip)) |  
                                (nivtip > 20 & c_modalidad1 != lag(c_modalidad1, default =  
first(c_modalidad1))), 1, 0)))
```

#Contemos n_unidades_de_servicio

#Volvemos a ordenar pero agregando cueanexo

```
Padrón_OA <- Padrón_OA %>%  
  arrange(cueanexo, nivtip, c_modalidad1) %>%  
  mutate(total_us = ifelse(row_number() == 1, 1,  
                           ifelse(cueanexo != lag(cueanexo, default = first(cueanexo)) |  
                                nivtip != lag(nivtip, default = first(nivtip)) |  
                                (nivtip > 20 & c_modalidad1 != lag(c_modalidad1, default =  
first(c_modalidad1))), 1, 0)))
```

#Contemos n_edificios

#Volvemos a ordenar por las variables que necesitamos

```
Padrón_OA <- Padrón_OA %>%  
  mutate(cui = as.numeric(cui)) %>%  
  arrange(cui, cueanexo) %>%  
  mutate(total_edificios = ifelse(row_number() == 1, 1,  
                                  ifelse(cui !=  
                                         lag(cui, default = first(cui)), 1, 0)))
```

#Volvemos a ordenar para que no queden desordenadas las bases definitivas

```
Padrón_OA <- Padrón_OA %>%  
  arrange(cueanexo, c_modalidad1, c_oferta)
```

#Guardamos la base en xlsx.

```
Padrón_oferta_activa_31_03_2025 <- Padrón_OA %>%  
dplyr::select(estado_est,  
  estado_est_desc,  
  sector,  
  sector_desc,  
  cue,  
  anexo,  
  cueanexo,  
  cui,  
  sede,  
  sede_admin,  
  estado_loc,  
  estado_loc_desc,  
  email,  
  nombre_est,  
  nombre_loc,  
  barrio,  
  calle,  
  num,  
  point_x,  
  point_y,  
  de,  
  comuna,  
  mingob,  
  d_mingob,  
  depfun,  
  d_depfun,  
  tipest,  
  d_tipest,  
  claverama,  
  c_oferta,  
  d_oferta_CABA,  
  c_modalidad1,  
  d_modalidad_basica,  
  ci_nivel,  
  ci_submodal,
```

```
codigo_oferta_CABA,  
total_establecimientos,  
total_localizaciones,  
total_edificios,  
total_ue,  
total_us) %>%  
dplyr::filter(estado_est == 1 & estado_loc == 1)  
  
write.xlsx(Padrón_oferta_activa_31_03_2025, "Nombre_de_archivo.xlsx")
```